**AS – 2 BIMESTRE**

**MATHEUS HENRIQUE BUTKOSKI SILVA**

**Tipos de arquiteturas, aplicáveis para desenvolvimento web**

Definimos como arquitetura as técnicas e os padrões usados para projetar e desenvolver uma aplicação. Ela fornece um roadmap e práticas recomendadas a serem seguidos durante esse desenvolvimento para que você tenha uma aplicação bem-estruturada.

Os serviços de front-end e back-end fazem parte da arquitetura da aplicação. O desenvolvimento de front-end tem como foco a experiência do usuário da aplicação, já o de back-end se concentra em fornecer acesso aos dados, serviços e outros sistemas existentes responsáveis pelo funcionamento dela.

Atualmente, o uso do JavaScript com HTML e CSS é uma das linguagens de programação mais populares para desenvolvimento de aplicação web.

**Vantagens/Desvantagens na utilização de frameworks em aplicações web**

Um framework serve para auxiliar o programador ou programadora na hora de construir uma aplicação web que já foi amplamente utilizada por outros especialistas.

Por ser um pacote de códigos padronizado, para utilizar um framework é necessário estar de acordo com o objetivo para o qual ele foi construído.

Por exemplo: existem frameworks específicos para o desenvolvimento de aplicativos, para a manipulação de bancos de dados, para a programação de games, para o desenvolvimento de redes sociais e etc.

Da mesma maneira, os frameworks são desenvolvidos a partir de diferentes linguagens de programação e, na hora de escolher um framework, é necessário estar de acordo com a sua linguagem nativa, ou a comunicação entre ele e a sua aplicação não ocorrerá.

As vantagens de utilizar um framework são várias:

**-Aumento de Produtividade;**

**-Código sem erros;**

**-Fácil Manutenção;**

**-Maior Segurança;**

Desvantagens de utilizar um framework:

**-Dificuldade em configurar;**

**-Mais códigos que o necessário em alguns projetos;**

Os frameworks mais famosos são:

**-React**

**-jQuery**

**-Bootstrap**

**-Django**

**-Vue**

**Bootstrap e seus componentes**

O Bootstrap é um framework front-end gratuito e de código-fonte aberto, que disponibiliza componentes prontos para você utilizar em seu projeto. Com ele, você consegue criar e personalizar sites responsivos para dispositivos móveis, desktops e notebooks, com componentes pré-construídos e com plugins JavaScript poderosos. Tudo isso porque toda a estrutura do HTML, CSS e Javascript já se encontram prontas, bastando apenas que o desenvolvedor importe suas dependências em seu projeto.

Outra funcionalidade interessante do framework é a quantidade de componentes que podem ser utilizados para proporcionar maior interatividade e melhorar a comunicação com o usuário. Confira, a seguir, alguns dos principais componentes do Bootstrap.

**Alerts**

O Bootstrap possibilita a configuração de forma simples e rápida de diferentes tipos de alertas, com cores específicas, de acordo com a situação. Para mostrar um alerta ao usuário para indicar atenção, por exemplo, basta utilizar a classe .alert-danger e será exibido uma caixa de texto com o fundo vermelho.

**Carousel**

Um componente muito utilizado no Bootstrap é o Carousel. Trata-se de um slideshow, ou seja, uma ferramenta que permite a exibição de imagens de forma responsiva. Ele também possibilita a inclusão de efeitos especiais para a transição da imagem e controles de exibição, como os indicadores de próximo e anterior.

**Navbar**

Outro poderoso componente do framework é o Navbar, que permite a construção de um sistema de navegação responsivo. É possível configurar diferentes formas de apresentação do menu, como escolher entre o posicionamento lateral ou superior, e definir a forma de exibição, que pode ser estendida ou contraída. Também é possível determinar a forma de exibição dos links do menu, que pode ser em formato de botão, link, menu suspenso, entre outras configurações, para facilitar a implementação da navegação do site.

**Frameworks em Javascript (jQuery, VueJS, ReactJS, AngularJS...) e sua utilização;**

Frameworks fornecem aos desenvolvedores a base necessária para construir aplicativos JavaScript. Isso economiza os desenvolvedores o esforço de começar do zero, utilizando uma base funcional para fazer as coisas funcionarem.

No caso do JavaScript, esta base inclui uma coleção de bibliotecas de código. As bibliotecas compilam o código que extrai funcionalidades específicas para o tipo específico de aplicativo em que você pode estar trabalhando. Em essência, a estrutura definirá a estrutura de todo o aplicativo.

Com as bibliotecas JavaScript, os desenvolvedores chamam funções específicas do código pai. Mas os desenvolvedores não podem chamar frameworks JavaScript. Em vez disso, os frameworks chamam e usam um trecho de código de uma maneira predeterminada. Frameworks definem o design do aplicativo.

A adoção de código de estruturas é mais holística do que o uso de bibliotecas. Uma estrutura não oferece apenas uma solução única, mas um modelo completo para desenvolvimento. A vantagem é mais organização e maior eficiência.

**As estruturas de JavaScript mais populares são:**

**Node.js**

Node.js não é exatamente uma estrutura JavaScript; é um ambiente de tempo de execução. Embora o JavaScript possa ser escrito diretamente no navegador da web, isso nem sempre é desejável.

É por isso que o Node.js oferece a capacidade para ferramentas de linha de comando e scripts do lado do servidor.

Embora o JavaScript geralmente opere no lado do cliente ou no front-end, o script do lado do servidor gera tempos de carregamento mais rápidos, pois a tecnologia do navegador não é necessária. Isso pode diminuir a frustração do usuário e aumentar o SEO.

**Vue.js**

O Vue.js se autodenomina a estrutura JavaScript “progressiva”. O nome deriva de sua filosofia de adoção incremental. No Vue.js, a biblioteca central concentra-se na camada de visualização apenas, portanto, qualquer funcionalidade adicional deve ser adotada em incrementos.

A estrutura usa um padrão de arquitetura model-view-viewmodel (MVVM). Esse padrão separa a interface gráfica com o usuário (IU) - ou a visualização - da lógica de negócios do aplicativo - ou modelo. A camada viewmodel é um meio conversor que sincroniza dados.

**AngularJS**

O AngularJS é mantido pelo Google e trata de complicações comuns na criação de aplicativos de página única (SPAs). Essa estrutura funciona aproveitando o vocabulário HTML em páginas da web dinâmicas.

No passado, o HTML só podia ser usado para conteúdo estático. Os SPAs funcionam carregando dinamicamente o conteúdo do servidor da web em vez do navegador da web. Como resultado, os SPAs funcionam de maneira semelhante aos aplicativos móveis e não precisam ser recarregados.

**Ember.js**

Ember.js é uma estrutura JavaScript que emprega um padrão de serviço de componente. Em oposição à arquitetura tradicional de model-view-controller (MVC), os componentes em Ember.js são centrais para a estrutura. Quase tudo em Ember.js pode ser categorizado como um serviço ou componente.

Os componentes são transitórios e manipulam o texto de marcação e os estilos de uma IU do aplicativo. Serviços são objetos que vivem durante o período de um aplicativo. Eles podem ser disponibilizados para diferentes partes de seus aplicativos e são mais bem usados ​​para estados persistentes.

**React**

React é uma estrutura JavaScript desenvolvida pelo Facebook que simplifica o processo de construção de UIs interativas. É a base do React Native, uma estrutura adjacente para a construção de aplicativos móveis.

Ambas as estruturas têm um fluxo de dados unilateral, considerado mais intuitivo do que a vinculação de dados bidirecional. O hot reload é outro recurso popular das estruturas React, permitindo que os desenvolvedores vejam imediatamente as alterações à medida que são aplicadas.

**jQuery**

Por ser de código aberto, o acesso à biblioteca jQuery é gratuito e sua base é constantemente incrementada por uma comunidade de desenvolvedores do mundo todo, que adicionam plugins, scripts e extensões para tornar os códigos cada vez mais funcionais. Por isso, a biblioteca jQuery pode ser usada em projetos de cunho pessoal e profissional, sem restrição.

O objetivo do jQuery é o incremento das linhas de código da linguagem de programação JavaScript, mas não de forma a adicionar complexidade. A biblioteca, pelo contrário, simplifica o que teria que ser escrito com strings, linhas e mais linhas de código em JavaScript.

Um simples efeito em uma página HTML que demandaria um complexo código em JavaScript pode ser substituído por única linha na sintaxe jQuery. Essa simplicidade de aplicação faz com que a biblioteca seja preferida por programadores experientes, mas também por designers e desenvolvedores com pouco conhecimento de programação.

**Diferenças em aplicações de Back/Front End;**

**Front-end**

O desenvolvedor front-end é responsável por “dar vida” à interface. Trabalha com a parte da aplicação que interage diretamente com o usuário. Por isso, é importante que esse desenvolvedor também se preocupe com a experiência do usuário.

Na parte de estudo, este profissional foca em HTML (linguagem de marcação), CSS (linguagem de estilo) e JavaScript (linguagem de script/programação).

É comum alguns profissionais de front-end trabalharem mais com a parte “criativa” e “artística” da aplicação. Assim como também há os que possuem um perfil mais voltado ao back-end, mesmo atuando como front-end. Esses possuem mais familiaridade com programação e suas nuances. Um tipo de perfil não invalida o outro.

**Back-end**

Como o nome sugere, o desenvolvedor back-end trabalha na parte de “trás” da aplicação. Ele é o responsável, em termos gerais, pela implementação da regra de negócio.

Em uma aplicação web, este desenvolvedor, quando focado, não toca na parte visual da aplicação. Por lidar com a regra de negócio, as vezes um programador de sistemas, como de aplicações comerciais e até científicas, pode ser chamado de desenvolvedor back-end. E geralmente, nessas aplicações, este desenvolvedor trabalha um pouco com a parte visual. Por isso, para este artigo, o desenvolvedor back-end levado em conta, é o desenvolvedor de aplicações web.

Quando falamos de back-end em desenvolvimento web, nos deparamos com várias linguagens, como Go, Clojure, C#, PHP, Java, Python, Ruby, entre outras. Cada uma possui vantagens e desvantagens em relação ao uso no desenvolvimento web, bem como no mercado de trabalho.

**BIBLIOGRAFIA**

<https://www.redhat.com/pt-br/topics/cloud-native-apps/what-is-an-application-architecture>

<https://kenzie.com.br/blog/framework/>

<https://rockcontent.com/br/blog/bootstrap/>

<https://www.treinaweb.com.br/blog/o-que-e-front-end-e-back-end>

<https://trio.dev/blog/javascript-framework>